

Распределённые алгоритмы

mk.cs.msu.ru → Лекционные курсы → Распределённые алгоритмы

Блок 44

Задача выявления конца вычислений

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2025, февраль–май

Завершение вычисления р.с. можно объяснить для кода каждого узла одной из двух причин:

- ▶ **Явное завершение узла**: узел выполнил весь свой код, больше нечего выполнять
- ▶ **Неявное завершение узла**: узел может продолжить выполнять свой код, но только в таких условиях, которые никогда больше не возникнут в вычислении

Например, в **алгоритме Ле-Ланна** избрания лидера в кольце последователь завершается неявно, «застревая» на команде приёма сообщения в бесконечном цикле из команд

$\text{receive}_{\text{prev}_p}(\text{pack}, v)$ $\text{send}_{\text{next}_p}(\text{pack}, v)$ lost

Неявное завершение вносит неопределённость в «высокоуровневое» поведение узла: неясно, выполнилась ли текущая задача и можно ли перейти к следующей

Чтобы избежать такой неопределённости, следует уметь определять, правда ли, что все узлы в системе завершились (явно или неявно)

Задача выявления конца вычислений (ВКВ) состоит в том, чтобы определить в каком-либо узле, достигнута ли заключительная конфигурация системы, с тем чтобы вслед за этим можно было оповестить систему о завершении выполнения всех узлов

Для такого оповещения можно использовать, например, волновой алгоритм или PIF-алгоритм, поэтому сконцентрируемся целиком на определении того, достигнута ли заключительная конфигурация

Как и для задачи сохранения снимка, решение задачи ВКВ — это **контрольный алгоритм**, который надстраивается над произвольно заданным **базовым алгоритмом**

После такой надстройки любую конфигурацию, в которой не допустимо ни одно базовое действие, будем называть **целевой** и просто **целью**

\mathcal{F} — так будем обозначать множество всех целей

Алгоритм ВКВ — это способ надстройки контрольного р.а. над произвольным базовым с соблюдением трёх требований:

1. **Прозрачность**: контрольный алгоритм не влияет на возможности выполнения базовых действий
2. **Живость**: после достижения \mathcal{F} алгоритм рано или поздно (спустя конечное время) начнёт оповещение
 - ▶ Начало оповещения будем отмечать командой `announce`
3. **Своевременность**: если в конфигурации γ начинается оповещение, то $\gamma \in \mathcal{F}$

Утверждение. Если р.а. прозрачен, то свойство \mathcal{F} монотонно

Значит, алгоритм ВКВ можно устроить, **например**, так:

- ▶ Время от времени собирать в узле снимок сети при помощи алгоритма сохранения снимка и INF-алгоритма
- ▶ Проверять по собранному снимку, достигнута ли целевая конфигурация

Но хотелось бы иметь и другие (более эффективные) средства выявления конца вычислений

При обсуждении задачи ВКВ будем использовать следующие допущения:

1. Если в канале $p \rightarrow q$ содержится базовое сообщение, то в q допустимо действие приёма этого сообщения
 - ▶ Это ограничение можно понимать так: рассматриваются базовые р.а., модифицированные так, чтобы все сообщения принимались узлом и сохранялись как часть состояния
 - ▶ Тогда команду `receive` можно понимать как соответствующую обработку сохранённого сообщения
2. Допустимость внутренних действий и действий отправки узла зависит только от состояния этого узла
 - ▶ Это ограничение обычно соблюдается на практике (особенно если верен пункт 1) и соблюдено во всех конкретных р.а., обсуждавшихся в курсе
 - ▶ Основываясь на этом ограничении, будем дальше говорить «действие узла **допустимо в состоянии** этого узла», имея в виду допустимость в любой конфигурации с этим состоянием узла

Если в состоянии узла допустимо хотя бы одно базовое внутреннее действие или базовое действие отправки этого узла, то будем называть это состояние **активным**, а иначе — **пассивным**

Узел назовём **активным**, если он находится в активном состоянии, и **пассивным** иначе

Активность узла означает, что базовый алгоритм в этом узле ещё не завершился

Пассивность узла означает, что базовый алгоритм в этом узле

- ▶ либо неявно завершился,
- ▶ либо рано или поздно выполнит действие приёма сообщения и продолжит выполняться, а до тех пор не выполнит ни одного действия

При обсуждении задачи ВКВ будем использовать следующие допущения:

3. Выполнив базовое внутреннее действие, узел становится пассивным, то есть
 - ▶ рассматриваются базовые р.а., в которых невозможно бесконечно выполнять одни только внутренние действия, и
 - ▶ базовые внутренние действия «укрупнены» насколько возможно
4. Сделать активный узел пассивным может только выполнение внутреннего действия
 - ▶ Если это не так, то можно добавить в соответствующем месте узла «фиктивное» внутреннее действие
 - ▶ В частности, узел всегда активен после приёма сообщения
5. В каждом узле содержится булева переменная $active_p$, которая имеет значение t , если узел активен, и f , если пассивен
 - ▶ На самом деле это либо контрольная переменная (и контрольный алгоритм обновляет её после каждого базового действия), либо булево выражение, подходящим образом разделяющее состояния

Теорема (о целевых конфигурациях). Конфигурация γ является целевой \Leftrightarrow в γ все узлы пассивны и в коммуникационной подсистеме (M) не содержится ни одного базового сообщения

Доказательство.

(\Rightarrow) Пусть $\gamma \in \mathcal{F}$

Если в γ хотя бы один узел r активен, то это (по определению) означает, что в r допустимо некоторое базовое действие — чего не может быть по определению \mathcal{F}

Если в M содержится хотя бы одно сообщение m , отправленное некоторому узлу r , то по допущению 2 в r допустимо действие приёма m — чего не может быть по определению \mathcal{F}

(\Leftarrow) Пусть $\gamma \notin \mathcal{F}$

Тогда в γ допустимо хотя бы одно базовое действие α некоторого узла r

Если α — внутреннее действие или действие отправки, то узел r активен

Если α — действие приёма, то в M содержится хотя бы одно сообщение, отправленное узлу r ▼

Теорема 1 о сложности ВКВ. В любом алгоритме ВКВ в худшем случае отправляется не меньше контрольных сообщений, чем в наилучшем волновом алгоритме для того же класса топологий, в котором все узлы являются инициаторами

Доказательство.

Рассмотрим базовый алгоритм, в котором каждый узел выполняет одно внутреннее действие и завершается

Тогда алгоритм ВКВ — это SYN-алгоритм:

- ▶ Обязательно следует выполнить действие $\beta = \text{announce}$
- ▶ β должно быть выполнено после базовых действий α_p всех узлов p

Известно (см. семинар 7), что SYN-алгоритм можно трактовать как волновой алгоритм, в котором $\beta = \text{decide}$

При этом в таком SYN-алгоритме каждый узел может выполнить α , а значит, является инициатором ▼

Теорема 2 о сложности ВКВ (Д.з. 1, непростое). В любом алгоритме ВКВ в худшем случае отправляется не меньше контрольных сообщений, чем базовых

Далее будем полагать, что алгоритм ВКВ «перехватывает» все базовые действия, то есть если в базовом алгоритме должно выполниться действие α , то в контрольном:

- ▶ Вслед за α выполняются подходящие контрольные действия (β)
- ▶ Если α — приём/отправка сообщения, то в β известны это сообщение и отправитель/адресат
- ▶ Действия β конечны, и ими не изменяются базовые элементы состояния

Поэтому, в частности, в описании алгоритма ВКВ все процедуры, кроме трёх, будут снабжены предусловиями, а эти три процедуры будут иметь заданное время начала выполнения:

1. $S_p(m, q)$ — контрольные действия узла p при отправке сообщения m узлу q
2. $R_p(m, q)$ — контрольные действия узла p при приёме сообщения m от узла q
3. I_p — контрольные действия узла p при выполнении базового действия